CSE 241 Algorithms and Data Structures **Fall Semester, 2005**

Midterm Exam

NAME: *October 20*

Your answers to each question should fit in the provided space. There is no need to repeat material from lecture/book (e.g. standard time complexity analysis of data structure methods), If you do need any extra space or scratch paper, some is available at the front of the class. If you have any questions about a problem, quietly come to the front. To pace yourself, you should allow no more than 1 minute/point. For example, on a 10 point problem, don't spend more than 10 minutes. Good Luck.

1. (15 points) Give tight asymptotic bounds for $T(n)$ in each of the following recurrences (i.e. it's sufficient to use Θ notation so use the master method whenever you can). You can assume that $T(1) = \Theta(1)$. As part of showing your work, you are required to give the value of "ℓ" and "k". You are welcome to do the rest in your head if you want but the more you show the more we can give partial credit if you made a mistake.

   (a) $T(n) = 3T(n/2) + 10 \ln n + 20n^2$

   (b) $T(n) = 2T(n/4) + 5\sqrt{n} + \log_{10} n$

   (c) $T(n) = 3T(n/3) + 5n(\log_2 n)^2 + \sqrt{n}$

   (d) $T(n) = 2T(2n/3) + n(\ln n)^3 + \frac{1}{2}n^2$

   Take each of the four answers above and put each one in one of the blanks below so that the resulting statement is true.

   $$\_\_\_\_\_ = \Omega(\_\_\_\_\_) = \Omega(\_\_\_\_\_) = \Omega(\_\_\_\_\_)$$

2. (5 points) Consider the following list of 3-letter initials

   CWA
   HWA
   JKK $\Longrightarrow$
   HCA
   CWJ
   ACA

   Illustrate the execution of radix sort on the above list of initials by by showing the list after each of the three phases of counting sort has completed.

3. (10 points) Complete the following divide-and-conquer algorithm to determine if all integers in an array a are equal. The initial call would be `allEqual(a,0,a.length-1)`. (Yes, there is an easy iterative algorithm for this problem. The goal here is to show me you can complete this divide-and-conquer solution and analyze it).

```
boolean allEqual(int a[],int p,int r){
   if (p == r)
      return true;
   if (A[p] != A[r])
      return false;









}
```

   Write a recurrence relation for your algorithm and then solve it to obtain the worst-case asymptotic time complexity for your algorithm.

4. (10 points) In this problem you will compute the expected number of comparisons made by the following algorithm to determine if a set of n integers are all equal.

```
boolean allEqual(int a[]) {
  for (int i = 0; i < a.length-1; i++)
     if (a[i] != a[i+1])
        return false;
  return true;
}
```

Compute the expected number of times the conditional (`a[i] != a[i+1]`) is executed when $n = 5$ (i.e. the array contains 5 elements) that are each equally likely to be a 0 or a 1.

Note: The probability that a sequence of x randomly select integers that are equally likely to be a 0 or 1 are the same is $(1/2)^{x-1}$.

Show your work to enough detail that we can figure out how you computed the answer.

5. You have been commissioned to write a program for the next version of electronic voting software for Dade County Florida. The input will be the number of candidates, d, and an array `votes` of size v holding the votes in the order they were cast where each vote is an integer from 1 to d. (You can assume that d is typically much smaller than v.) The goal is to determine if there is a candidate with a majority of the votes. If there is a candidate with the majority of the votes, you are to output the indices in `votes` for the elements that hold a vote for the winning candidate.

   (a) (15 points) Describe the algorithm that you would recommend to solve this problem. Analyze the time complexity of your algorithm (as a function of d and v) and very briefly argue why it was the best choice.

(b) (10 points) Consider a model of computation in which you can only access `votes` by asking if `v[i] == v[j]` for any i, j. Using the decision tree lower bound technique give the best lower bound you can on the number of comparisons that must be made to solve this problem under this model of computation when there are 2 candidates and 4 voters. Note that at least 3 votes are needed here for a majority.

6. (20 points) You are given a set S of w words to use in a word list to be used by a spell checker. You are also given an n word document. Your task is to list all words in the document that are not in S. Give an $O(n + w)$ expected time algorithm that only requires space for at most $2w$ references (plus the space needed to store the words in S and the words in the document).

   Be sure to analyze the time and space complexity of your solution. You need only include enough detail to argue that your solution satisfies the stated goals.

Problem	Points Possible	Points Received
1	15	
2	5	
3	10	
4	10	
5a	15	
5b	10	
6	20	
total	85	